A Goal-based Method for Automatic Generation of Analytic Needs

Ben Abdallah Mounira, Zaaboub Haddar Nahla and Ben-Abdallah Hanene
MIR@CL Laboratory, University of Sfax, Tunisia
{Mounira.Benabdallah, Nahla.Haddar, Hanene.Benabdallah}@fsegs.rnu.tn

Keywords: Analytical Requirements Generation, Goal Requirements Language (Grl), Uml Modeling.

Abstract: In this paper, we are interested in the requirements engineering of decision support systems. In particular, we propose a method, called *Analytic Requirements Generation Method* (ARGeM), for automatic generation of analytic requirements. Our method meets the strategic goals of the enterprise and produces loadable DW schemas. It begins with modeling the goals of the enterprise and uses the UML IS modeling artifacts to generate automatically a complete set of candidate analytic needs. These needs are, subsequently validated by the decision makers who are thus directly involved in the specification process. Once validated, needs contribute to the design of the DW.

1 INTRODUCTION

For decision making, analytic needs that must be satisfied by the system's data warehouse (DW) are specified during the analysis phase. As one of the early stages of system development, this phase implies major problems, if it is inaccurate or incomplete and does not meet the entire user's needs. Thus, it should attract special attention and must be fully supported by effective methods.

On the other hand, several surveys indicate that a significant percentage of DWs fail to achieve the business goals or are spectacular failures. One reason for this is that the requirements analysis is typically overlooked in real projects (Giorgini et al., 2008). Thus, this stage should be based on a goal oriented framework for requirements engineering as the DW aims at providing adequate information to support decision making and to achieve the goals of the organization.

Moreover, existing approaches to decision system development, such as (Golfarelli et al., 1998) (Moody et al., 2000) (Giorgini 2008) works, always consider that the information system (IS) of the enterprise is already computerized and operational for a large period. Therefore, these approaches often encounter some problems such as the lack of source schemas. Moreover, the lack of a decision support system aligned with the IS since its implementation, can threaten its survival. To remedy these problems, it is important to have a decision support built at the same time as the IS and constantly aligned with it.

This work proposes a method, called *Analytic Requirements Generation Method* (ARGeM for short), for automatic generation of analytic requirements that meet the strategic goals of the enterprise and produce loadable DW schemas. Our method begins with modeling the goals of the enterprise and uses the UML IS modeling artifacts to generate automatically a complete set of candidate analytic needs. (The use of UML is due to the fact that this language is a defacto standard for IS modeling). These needs are, subsequently validated by the decision makers who are thus directly involved in the specification process. Once validated, these needs contribute to the DW design.

In the following, Section 2 gives a state of the art of works on the analytic requirements engineering. Section 3 presents our approach to generate analytic needs. The last Section concludes this work and discusses its prospects.

2 RELATED WORKS ON ANALYTIC REQUIREMENTS ENGINEERING

Although most DW design methods claim that there must be a phase devoted to analyze the requirements of an organization (Golfarelli et al., 1998) (Kimball 2002) (Lujan-Mora et al., 2006), this phase does not generate the same interest in both types of DW design approaches: bottom-up and top-down. Indeed,

Mounira B., Nahla Z. and Ben-abdallah H.
A Goal-based Method for Automatic Generation of Analytic Needs.
DOI: 10.5220/0004462201500156
In *Proceedings of the Second International Symposium on Business Modeling and Software Design (BMSD 2012)*, pages 150-156
ISBN: 978-989-8565-26-6

bottom-up approaches start from a detailed analysis of the data sources (Golfarelli et al., 1998) (Moody et al., 2000). Analytical needs are expressed directly by the designer who must select relevant blocks of data to decision making and determine their structuring according to the multidimensional model (Golfarelli et al., 1998) (Moody et al., 2000) (Cabibbo et al., 1998) (Prat et al, 2006).

Therefore, these approaches assume that decision makers have a good knowledge about the models of operational data, and a perfect understanding of the structures of the data source. Thus, they marginalize the analysis phase of the OLAP requirements in a decision system design. Therefore, the DW may not satisfy all its future users, and may, therefore, probably fail (Giorgini et al., 2008). In addition, all these approaches produce multidimensional schemas regardless of the needs of the decision makers. Thus, the produced schemas are far from covering the goals of the organization.

Unlike bottom-up approaches, top-down ones start by determining information needs of the DW users. These approaches collect and specify the user requirements using different formalisms: goal based models, UML use cases, query languages or decision oriented models. The problem of matching user requirements with the available data sources is treated only a posteriori.

Most goal based approaches are essentially founded on the conceptual framework i* (Giorgini et al., 2008) (Zepeda et al., 2008) (Franch et al., 2011). Requirements' specification is carried out manually from diagrams modeling the enterprise and its goals. Thus, these approaches may overlook some requirements as they may specify needs not covered by the sources. Moreover, they do not directly involve the decision maker. Besides, i* is not a standard and does not provide all the concepts necessary for modeling purposes. So, it requires specific training and tools that support it in order to be used.

The use case (UC) based approaches adopt the UCs of UML to represent the analytic needs (Luján-Mora 2006) (Shiefer et al., 2002). Thus, because of the absence of a precise oriented decision syntax for enouncing UC actions, it becomes very difficult, to identify potential decision elements from the specification. Moreover, the fact that the UML does not model the organization goals, using UC cannot guarantee the coverage of all enterprise goals.

The query based approaches are the most used in literature (Romero et al., 2006) (Bargui et al., 2008), because queries expressed in natural language or pseudo language are easy to understand by decision makers. However, the non-exploitation of the source information impedes obtaining, from the beginning, an optimal set of analytic needs.

In the decision based approaches, needs are specified using decision concepts expressed in a given formalism (Kimball 2002) (Golfarelli et al., 1998). Although the models used by these approaches are characterized by their decision orientation, they remain difficult to understand by decision makers who lack design expertise. Moreover, the fact that there is no well-defined framework for defining goals, the specified needs do not guarantee the achievement of these goals.

The overview presented above on top-down approaches reveals two important criticisms. First, none of the presented methods propose joint modeling of the DW and the IS. This may impede the alignment of the DW to the IS which in turn may produce unloadable schemas. In addition, it does not guarentee the completeness of the analytic needs. Second, specifying needs without linking them to their goals may not lead to the achievement of the expected goals.

To remedy these problems, we propose an analysis method called *ARGeM* (*Analytic Requirements Generation Method*) for automatic generation of analytic requirements that meet the strategic goals of the enterprise and produce loadable DW schemas. Our method begins with modeling the goals of the enterprise and uses the UML IS modeling artifacts to generate automatically a complete set of candidate analytic needs. The aligned modeling of the DW and the IS facilitates the co-evolution of both systems. Another advantage of our method is that it involves directly the decision makers in the specification of their needs by validating the generated requirements.

3 GOAL DRIVEN ANALYTIC REQUIREMENTS GENERATION METHOD

ARGeM consists of three steps (cf. Figure1): i) GRL model construction, ii) analytic element identification and iii) analytic requirements generation. In the following sub-sections, we detail these steps.

3.1 Construction of the GRL Model

Since achieving the qualitative goals of an enterprise is the main purpose behind modeling a DW, it is obvious to begin with determining these goals and taking them as a start point for deriving any analytic needs. Thus, the first step in our approach is to construct automatically a model representing the qualitative goals of the enterprise. As a pre-condition for this step, we suppose the existence of a business strategy definition for the enterprise, represented with one of the current models such as the ISO[1] model. This document, which defines all the strategic goals of the enterprise, usually exists since the establishment of an enterprise requires its presence. We also consider the use case model of the UML IS modeling documentation which gives the functionalities of the IS that helps to reach the strategic goals.

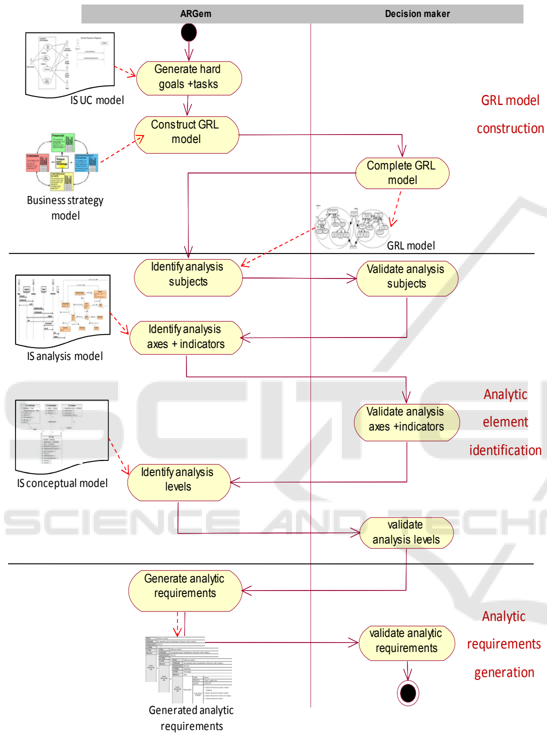

Figure1: The steps of ARGeM.

The product of the first step is a goal model represented with the standard goal requirements language GRL (ITU-T, 2008). With this language, it is possible to represent both functional (low level) goals and their performing tasks, and qualitative (high level) goals that the former tend to meet.

Furthermore, the functional goals of the enterprise are realized by its IS. Their specification is part of the SI design. Indeed, the UC model represents these goals as UCs and scenarios. Thus, it is possible to transform these latter into functional goals and tasks in the GRL model. Subsequently, this model will be completed by the high level goals and their dependencies with all other model elements basing on the business strategy model (BSM) and decision makers directives.

To ensure the generation of the goals from the UC model, we are inspired from the works of (Vicente 2009) (Cysneiros et al., 2003). Basing on these works, we define three rules for transforming the UC concepts into GRL.

Rule1: Transformation of UML Actors

Each UML actor can be transformed into a GRL actor that has the same name. A generalization/specialization relationship between two UML actors becomes an inclusion relationship between the two corresponding GRL actors.

In UML, UCs can be classified into three types: business, support and decision ones (Morley et al., 2008). A business UC describes a business activity while a support UC manages a system resource that is necessary for a business activity. A decision UC provides useful information for decision making. This classification is not standard in UML but can be easily carried out by stereotyping all UCs with one of the three stereotypes: “business”, “support” or “decision”.

UCs are described through nominal and alternative scenarios. In GRL, the goals of an enterprise are of two types: hard and soft. A hard goal is a low level goal that represents a state or a condition that the stakeholders would like to achieve in the enterprise. While a soft goal is a high level one and describes qualitative aspects rather than functional ones. The GRL goals are achieved by executing a set of activities called tasks (ITU-T 2008).

Rule2: Transformation of UCs

Business UCs become hard goals. The scenarios of a UC are potential tasks composing the corresponding hard goal. By contrast, a support UC becomes a GRL resource representing a physical or an information entity.

Rule 3: Transformation of Relationships between Actors and UCs

The communication relationship between an actor and a UC results in the placement of the corresponding goal in the GRL actor generated from the UML one.

Applying these three rules produces a GRL model containing all the elements modeling the enterprise goals except the soft ones. To complete

[1]International Organization for Standardization. http://www.iso.ch

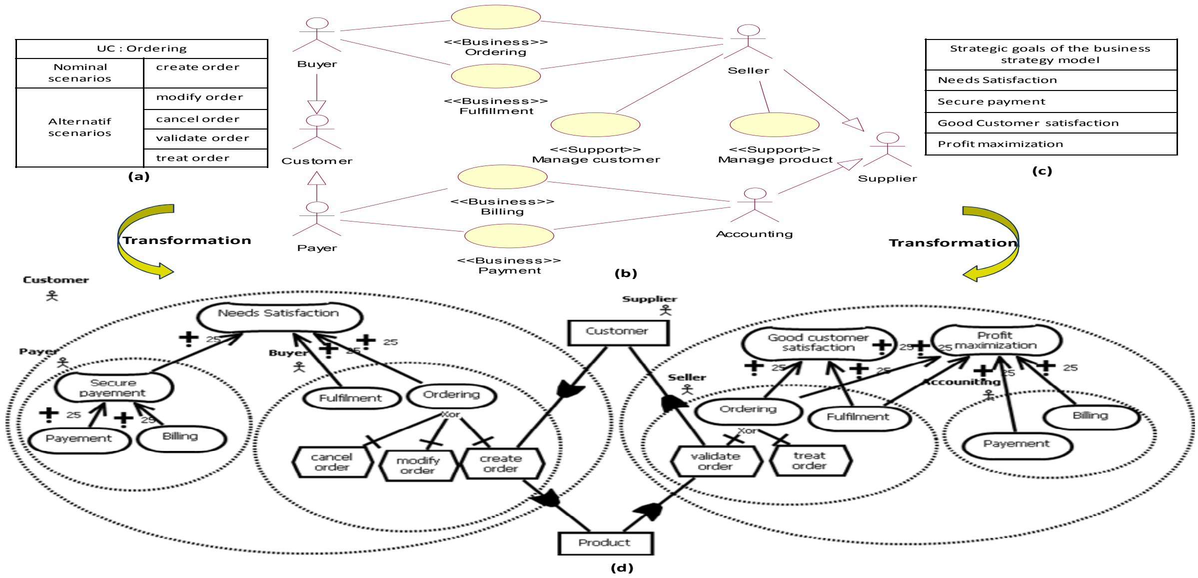

Figure 2: Extract of the GRL model (d) constructed from a UC model ((a) and (b)) and a BSM (c) from the "online sales".

this model, the soft goals specified in the BSMare automatically copied in the GRL one. Then, the missing relationships between soft and hard goals are added by the decision maker.

Figure 2 shows an extract of the GRL model (d) constructed from a UC model ((a) and (b)) and a BSM(c) from the "online sales" domain.

Since the analytic needs aim to analyze information from the IS in order to achieve the stated goals, we must identify the analytic elements from the IS (analysis subjects, analysis axes, indicators) that contribute to formulate these needs. To do this, in the second step of our method, we start from the UML IS modeling artifacts and the GRL model built in the first step. The relationship between the goals of the enterprise and the system functionalities made by the above defined three rules is used to identify these elements.

3.2 Identification of Analytic Elements

This step aims to identify the elements contributing to formulate analytic needs in terms of subjects, indicators, axes and the analysis levels.

3.2.1 Identification of Analysis Subjects

An analysis subject is an activity of the target functional system in order to achieve the company goals. Thus :

RS: each hard goal contributing to satisfy a soft goal is a potential analysis subject.

This rule is justified by the fact that a subject to be analyzed represents business missions. These latter are supported by business UCs. On the other hand, generating subjects (indirectly) from UCs guarantees the loading of these subjects from the source. From the GRL model of Fig 2, the rule RS identifies the subjects "Ordering", "Fulfillment", "Billing" and "Payment".

3.2.2 Identification of Analysis Indicators and Axes

Recall that a subject is formed of indicators and is analyzed from different perspectives called analysis axes representing the observing and the recording context of the indicators. Therefore, the identification of the indicators and the axes of an analysis subject which amounts to analyzing the corresponding business UC. This analysis takes into consideration all the artifacts related to the UC. In particular, we focus on the interaction diagrams (ID) describing the scenarios of the UC and on the class diagram. Since an ID describes the communication between objects that participate in the execution of the UC, this communication serves to identify the potential axes and indicators of the subject. Then, using the class diagram, we consolidate the identified elements and we determine the analysis levels of each axis.

To identify the analysis axes, we define the following rule:

RA: In the ID describing the scenario of creation of a business object corresponding to a subject *S*, let *A* the set of business objects created during this

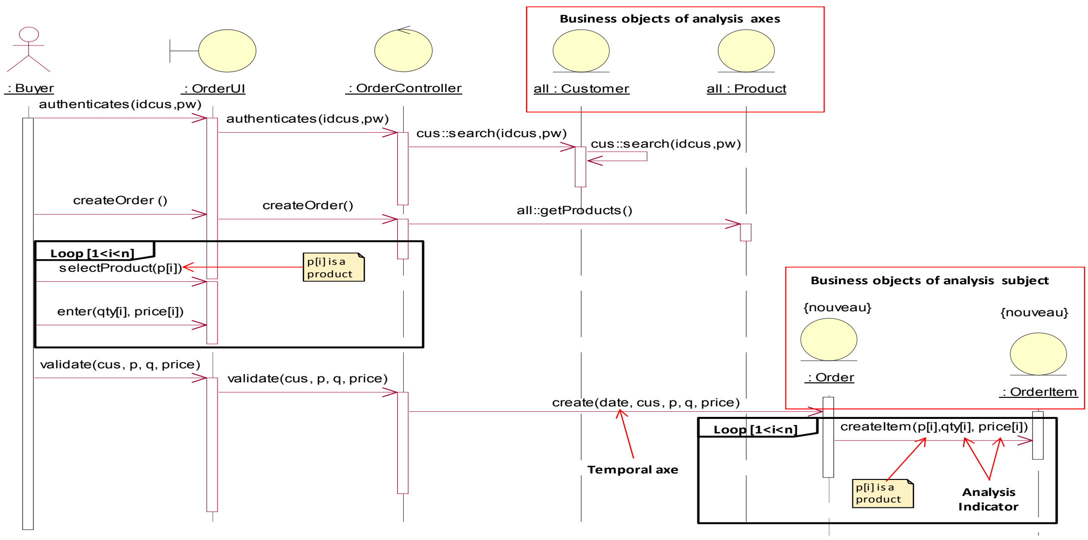

Figure 3: Identification of subject classes, analysis axes and indicators of the « Ordering » subject.

scenario. A business object that is involved in the scenario but does not belong to *A* is a potential axis for *S*. In fact, such an object provides required information for creating the objects of *A*. Moreover, a date parameter of a message having as destination an object belonging to A is a potential temporal axis for S.

It is rather interesting to mention here that a scenario of creation of a business object is easy to identify in UML because this language provides a different notation for the creation message.

Fig 3 illustrates the application of rule *RA* on the sequence diagram formalizing the scenario "Create order" of the "Ordering" UC. The business objects of type "Order" and "OrderItem" created in this scenario correspond to the "Ordering" subject. The business objects of type "Customer" and "Product" correspond to the analysis axes of this subject. The parameter "date" of the message "create" sent to the object "Order" represents a temporal axis for the analysis subject.

We identify indicators by defining the following rule:

RI: in the ID of the creation scenario of a business object corresponding to a subject *S*, a numerical parameter of a message having as destination an object corresponding to *S* and which does not refer to an axis represents a potential indicator for *S*. Indeed, such a parameter will be used in the creation of the destination object.

In the sequence diagram of Fig 3, *RI* produces the indicators "qty" and “price” for the subject "Ordering".

Identification of subjects within business UCs and IDs is more accurate than within class diagrams because pure structural information such as attribute types and multiplicity of relationships is not sufficient to decide of the relevance of a class as being a subject.

In contrast, functional information, provided by UCs, and dynamic information of IDs are more efficient for the subject classes’ detection. Indeed, a subject class constitutes a central class around which all interactions take place. Thus, it is easy to identify such a class in a business UC and in an ID.

To consolidate the results provided by rules *RS*, *RA* and *RI*, and to identify the analysis levels of each axis, we use the UML class diagram. To do this, we partition this diagram into *clusters*. Each cluster contains all classes representing a single analysis subject and all classes that are related to it directly or indirectly. Thus, we obtain as many clusters as analysis subjects. The goal of clustering is to facilitate the consolidation phase and, subsequently, to identify the analysis levels.

Recall, first, a primary rule of consistency between the class diagram and IDs: all communication between objects in a system must be supported by static relationships between their classes. Thus, regarding this rule, all classes in a cluster that are directly connected to those corresponding to the analysis subject correspond to axes.

In addition, to consolidate indicators, we define the following rule:

RI’: an indicator *m* identified for a subject *S* is an attribute of a class corresponding to *S*.

Figure 4 illustrates the consolidation of the indicators "qty" and "price" as attributes of the class "OrderItem" by applying the rule *RI'* in the class cluster of the "Ordering" analysis subject.

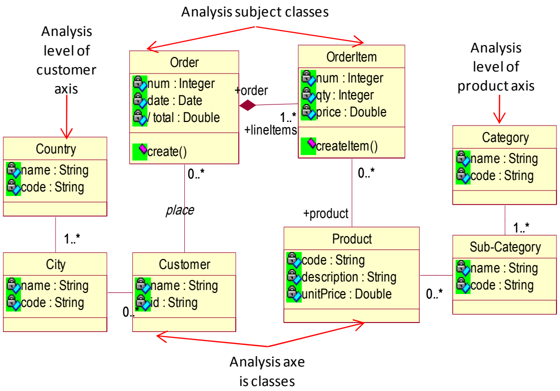

Figure 4: Identification of analysis levels of the product and the customer axes in the "Order" cluster.

3.2.3 Identification of Analysis Levels

Since a subject is analyzed according to different axes, each of which has one o many levels, then, these levels correspond to the attributes of the class axis and those of all classes that are related to it. Moreover, since the levels of an axis are generally non- numerical, we must examine the types of identified attributes.

In addition, during the OLAP process, data are usually analyzed starting from a low level detail to the most detailed one. To formulate analytic needs according to the OLAP process, ie, requirements that analyze a subject according to different levels of axes starting from the lowest to the most detailed one, it is crucial to determine the order of each level in an axis. So, to identify analysis levels, we define two rules *RN1* and *RN2*.

RN1: in a cluster, each class directly or indirectly connected to a class axis is a potential level for this axis. This level has the same range as the number of relationships that separates the class level to class axis. The name of this level is the same as the attribute playing the identifier role in the class level.

RN2: each non-numerical attribute of a class axis (class level) is a potential level for this axis (level). In particular, the levels of a temporal axis are the attributes composing a date such as year, month, and day. The decision maker can also add his own temporal levels.

Since our method does not automatically distinguish between a descriptive attribute and a level attribute, we expect to the decision maker to make this distinction.

In Fig 4, *RN1* identifies, for example, the levels "sub-category" and "category" for the axis "Product". *RN2* generates, for example, the analysis levels "id" and "name" of the axis "Customer".

3.3 Analytic Requirement Generation

For the specification of analytic requirements, we propose to use the template and the syntax proposed in (Bargui et al., 2008) as a means used by decision maker to express his needs. This template is instantiated with the analytic elements identified in the previous phase.

Figure 5 shows the analytic requirement of analyzing the performance of the "ordering" process for an online selling enterprise in order to maximize the profit.

TITLE	Order process analysis		
SUMMARY	This requirement analysis the performance of the process order according to....		
UPDATE DATE	11/04/2012		
ACTOR	Seller		
PROCESS	Order		
Soft Goal 1: Maximize profit	INDICATOR	LABEL	**Total amount**
		FORMULA	**qty * price**
		ANALYTIC QUERIES	1) Analyze the **total amount** by *sub-category* and *category* of a **product** according to *month* of a **date**. 2) Analyze the **total amount** by *city* and *country* of a **customer** according to *year* of a **date**. 3) Analyze the **total amount** by *code* and *description* of a **product** by *id* and *name* of a **customer** according to *month* and *year* of a **date**.

Figure 5: Extract of the generated analytic requirements for the process "Ordering".

4 CONCLUSIONS

In this paper, we proposed an analysis method for automatic generation of analytic requirements which meet the strategic goals of the enterprise and produce loadable DW schemas. Our method begins with modeling the goals of the enterprise and uses the UML IS modeling artifacts to generate automatically a complete set of candidate analytic needs.

The novelty of our method is the aligned modeling of the DW and the IS which facilitates the co-evolution of both systems. Another advantage of our method is that it involves directly the decision makers in the specification of their needs by validating the generated requirements.

As future work, we are examining how we can extract multidimensional concepts from generated requirements.

REFERENCES

Bargui, F., Feki, J., Ben-Abdallah, H., 2008. A natural language approach for data mart schema design. *9th International Arabic Conference on Information Technology (ACIT)*, Hammamet-Tunisia.

Cabibbo, L., Torlone, R., 1998. A logical Approach to Multidimensional Databases. *6th International conference on EDBT*, Valencia, Spain, LNCS 1377, pp. 183-197.

Cysneiros, G. A.A., Zisman, A., Spanoudakis, G., 2003. A Traceability Approach for i* and UML Models. *2nd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems - ICSE 2003*, Portland, May 2003.

Franch, X., Maté, A., Trujillo, J., Cares, C., 2011. On the joint use of i* with other modelling frameworks: A vision paper. *RE*, 133-142.

Giorgini, P., Rizzi, S., Garzetti, M. (2008). GRAnD: A Goal-Oriented Approach to Requirement Analysis in Data Warehouses. *Decision Support Systems (DSS) jounal*, Elsevier, Vol 45, Issue 1, pp. 4-21.

Golfarelli, M., Rizzi, S., 1998. A methodological framework for data warehouse design. *DOLAP*, pages 3–9.

ITU-T: International Telecommunications Union: Recommendation Z.151, 2008. *User Requirements Notation (URN) – Language definition*. Geneva, Switzerland.

Kimball R., 2002. *The Data Warehouse Toolkit*, Wiley, New York, 2nd edition.

Luján-Mora, S., Trujillo, J., Song, I. 2006. A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering*, 59(3), pp. 725-769.

Moody, D., Kortink, M., 2000. From enterprise models to dimensional models: A methodology for data warehouse and data mart design. *2nd DMDW*, Stockholm, Sweden.

Morley, C., Hugues, J., Leblanc, B. 2008. *UML 2 pour l'analyse d'un système d'information - Le cahier des charges du maître d'ouvrage*. Dunod, 4ème édition, Paris. ISBN 978-2100520978.

Prat, N., Akoka, J., Comyn-Wattiau I., 2006. "A UML-based data warehouse design method". *Decision Support Systems*, vol. 42, pp. 1449-1473.

Romero, O., Abelló, A., 2006. Multidimensional design by examples. *DaWaK*, LNCS 4081, pp. 85–94.

Shiefer, J., List, B., Bruckner, R. 2002. A holistic approach for managing requirements of data warehouse systems. *Americas Conference on Information Systems*.

Vicente, A.A., Santander, V.F.A., Castro, J.F.B., Freitas, I., Matus, F.G.R., 2009. JGOOSE: A Requirements Engineering Tool to Integrate I* Organizational Modeling with Use Cases In UML. *Ingeniare. Revista Chilena de Ingenier,* 17(1) 6-20.

Zepeda, L., Celma, M. , Zatarain, R. 2008. A Mixed Approach for Data Warehouse Conceptual Design with MDA, O. Gervasi et al. (Eds.), *ICCSA 2008*, Part II, LNCS 5073, pp. 1204–1217.